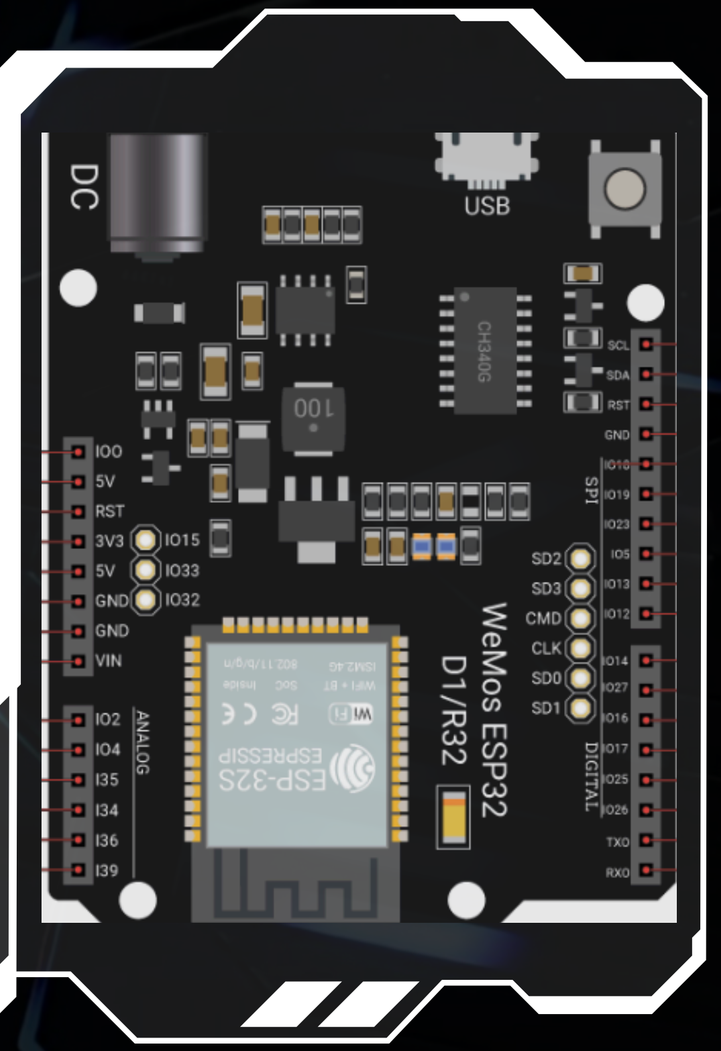
UNIVERSITÀ DEGLI STUDI DI SALERNO

SECURE IOT SENSOR NODE WITH ENCRYPTED COMMUNICATION

Using ESP8266, MQTT, AES, HMAC, and TLS



PROJECT **GOAL**

It consists of an ESP8266 board connected to a DHT11 sensor that reads environmental data.

The data is encrypted using AES and authenticated with HMAC before being sent securely through MQTT over TLS.

On the subscriber side, a Python program decrypts the data and verifies its integrity.
The main goal is to ensure confidentiality and integrity of IoT data communication.

SYSTEM **ARCHITECTURE**

DIAGRAM :

ESP8266 + DHT11 → TLS → MQTT BROKER → TLS → PYTHON SUBSCRIBER

- **PUBLISHER** = ESP8266 READS SENSOR DATA
- **BROKER** = MOSQUITTO (TLS-ENABLED)
- **SUBSCRIBER** = PYTHON APP DECRYPTS + VERIFIES

SECURITY **LAYERS**

Transport Security (TLS)

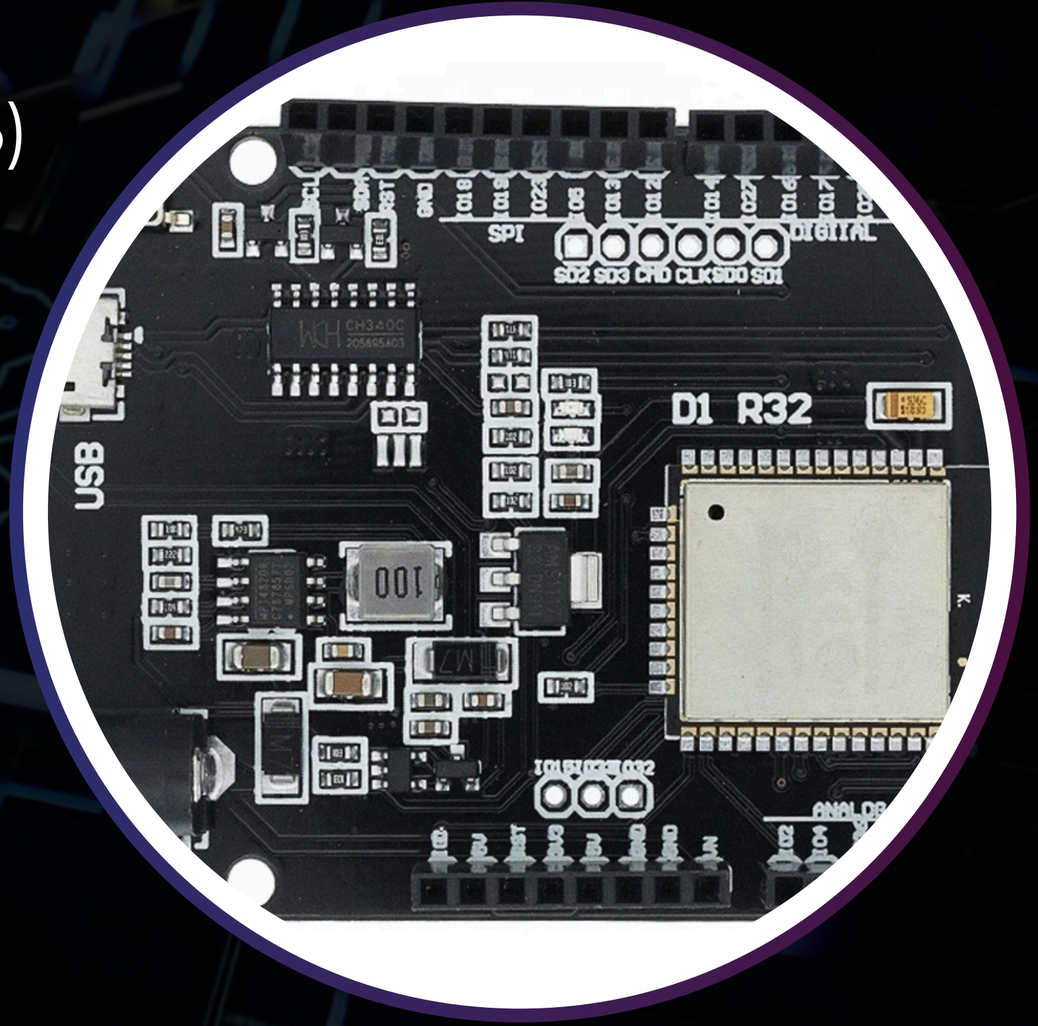
- Protects channel ESP ↔ Broker ↔ Subscriber
- TLS Encryption: Secures the channel between the device and broker

Application Security (inside payloads)

- AES-128-CBC encryption (confidentiality)
- HMAC-SHA256 (integrity, authenticity)
- Timestamp freshness (anti-replay)
- Base64 Encoding: makes encrypted data safe and easy to transmit as text for MQTT

PUBLISHER (ESP8266)

- **Hardware**: Wemos D1 Mini + DHT11
- **Libraries**: ESP8266WiFi, WiFiClientSecure, PubSubClient, ArduinoJson, Crypto, DHT
- **Steps per cycle**:
  - READ TEMP/HUMIDITY
  - BUILD JSON PAYLOAD
  - PAD + AES-128-CBC ENCRYPT (RANDOM IV)
  - COMPUTE HMAC-SHA256 OVER TS|IV|CT
  - PUBLISH ENCRYPTED JSON VIA MQTT



BROKER
MOSQUITTO TLS

- **TLS-enabled on port 8883**
- **Configured with** :
  - CA certificate
  - Server certificate
  - Private key
- **ESP8266 connects only if broker cert matches SHA1 fingerprint**
- **Python subscriber connects with CA certificate (tls_set)**

SUBSCRIBER **PYTHON**

- **Libraries: paho-mqtt, pycryptodome, python-dotenv**
- **Loads AES + HMAC keys from .env**

- **Process per message :**
  - Verify timestamp freshness (anti-replay)
  - Verify HMAC (integrity + authenticity)
  - Base64 decode IV + ciphertext
  - AES-128-CBC decrypt + unpad
  - Parse JSON (device_id, temp, hum)

EXAMPLE PAYLOADS

- **PLAINTEXT (ESP8266 BEFORE ENCRYPTION):**

  {"DEVICE_ID":"ESP8266-1","TS":1759152722,"TEMP":25.0,"HUM":46}

- **PUBLISHED ENCRYPTED PAYLOAD:**

  {"TS":1759152722,"IV":"...","CT":"...","HMAC":"..."}

- **SUBSCRIBER OUTPUT (DECRYPTED):**

  {"DEVICE_ID":"ESP8266-1","TS":1759152722,"TEMP":25.0,"HUM":46}

ACHIEVEMENTS

- Secure end-to-end IoT communication ✓
- Confidentiality → AES encryption ✓
- Integrity + Authenticity → HMAC-SHA256 ✓
- Replay Protection → timestamps ✓
- Transport Protection → TLS + fingerprint pinning ✓
- Successfully integrated ESP8266, Mosquitto, and Python subscriber ✓

FUTURE **WORK**

- **Security testing demos:**
  - Replay attack rejection
  - Tampered packet rejection

- **Data logging to DB (SQLite, InfluxDB)**
- **Visualization (Grafana, Matplotlib)**

UNIVERSITÀ DEGLI STUDI DI SALERNO

THANK YOU!